AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph bridging pages 7 and 8 with the following:

Figures 3 and 4 depicts the memory layout of the bedrock mapping mechanism. Since only those portions of the map, 26, which actually changed are updated, there's no contiguous, full-size "main" or "shadow" arrays; rather, sector-size groups of mapping entries serve as either, in accordance with a ping-pong bit flag in an upper-level mapping structure. The superblock, 20, contains 512 bytes = =4096 bit flags, each capable of controlling an underlying mapping page. Following the convention of grouping map entries into 512 byte segments, and representing block addresses as unsigned long 4 byte entries, one has 128 entries per map segment. Should each superblock bit control (switch) a segment directly, one would end up with only 4096 *128 -524288 bottomlevel blocks. Since one can't enlarge the superblock, 20, without violating the atomicity condition, and the map entries and their segment sizes are likewise fixed, the invention introduces an intermediate mapping layer, which is called pages, 22. Thus, the overall mapping becomes three-level:

superblock → pages → map segments

Please replace the paragraph bridging pages 9 and 10 and the first full paragraph on page 10 with the following paragraphs.

Figure 4 depicts the memory layout of the bedrock mapping mechanism 29 map[logical] = = physical. The sb, 30, and pp's, 35, are bit vectors: bit *sb*[i] chooses pp_0 v. pp_1 segment of 4096 bits, in the range governed by *i*; similarly bit pp[*j*] chooses map_0 v. map_1 segment of 128 unsigned long physical entries (32 bits each) for the index range (of logical addresses) based on *j*. The touched bit vectors, 37A-C, allow for selective writes of their "parents" upon commit or abort. The free physical, 40, and logical, 45, array lists (the latter sharing its array with map) allow for constant-time slot and address allocation, respectively; physical_freed_ head (pfh) protects slots freed in the current transaction from reuse until a

sync/abort.

The slalom has been abstracted and parameterized for three different occasions: read, write, and abort. It works in cooperation with the touched arrays of flags 37A-C, and is at the core of bedrock open, sync (commit), and abort operations. The touched flags 37 A-C, set naturally when allocating and overwriting blocks, provide for one to write out only those parts of the maps and pages that were actually updated. Similarly, an abort will reread only those sections of that were tinkered with and need restoration from disk. In addition, one can set all bits of all touched arrays 37 A-C before an "open", then simply make a slalom read, which consequently refills all the maps and the superblock 20. Also, since the touched arrays 37 A-C accurately record all of the bit flags changed, nothing need be read but the map 29 itself when aborting.

Please replace the first two paragraphs on page 11 with the following paragraphs:

As each page 22 is read, its governing sb touched bit 37A is reset.

3. Similarly, the pages 22 are now traversed, and if aborting, simply restored by reversing those bits set in pp_touched 37B (ping-pong pages touched bits). The maps should be read back from the disk when aborting, as the original modified slots are overwritten in memory with the shadow ones. The rest is analogous to the superblock traversal.